

Flash ActionScript Quick Reference

Author: Jialong He
Jialong_he@yahoo.com
<http://tiger.la.asu.edu>

Introduction

Using Flash to create animations on the web is popular because the flash player is installed on most computers and the published flash file (SWF file) is small. Flash has a powerful scripting language called ActionScript. You can use write script to manipulate and control objects on the stage. Its syntax is similar to JavaScript (or C++).

Note: this quick reference is based on ActionScript 2.0 language reference.

Script Example

A script can be associated with a keyframe or with an object. To test the following script, copy it to the action panel. Press "Ctrl + Enter" to start the flash file.

```
//=====
// A simple ActionScript
//=====
for (Cnt=1; Cnt<10; Cnt++) {
    trace(Math.random());
}
trace ("Hello, World!");
```

Operator

+, -, *, /, %	Add, Subtract, Multiply, Division, Remainder
+=, -=, *=, /=, %=	Combine with assignment operator
++, --, []	Increase, decrease, Array access
==, !=, <, <=, >, >=	Comparison, equal, not equal, less than, ...
!, &&, 	Logical NOT, AND, OR
<<, >>, >>>	Bit shift, left, right, right unsigned
~, &, , ^	Bitwise NOT, AND, OR, XOR
new, delete	Allocate (delete) an object
typeof, instanceof	get expression type, test an instance
//, /* */	One line and multiple line comments

Constants and Compiler Directives

true, false, undefined, null, NaN, Infinity, newline	Predefined constants
#initclip	Initialization actions are executed only once when a SWF file is played
statements(s)	
#endinitclip	
#include "filename.as"	Include external ActionScript

Program Flow Control

if (condition){ statement(s); } else { statement(s); }	if (age>=18) { trace("welcome, user"); } else { trace("sorry, junior"); }
for (init; condition; next) { statement(s); }	for (Cnt = 1; Cnt<10; Cnt++) { trace(Cnt); }
switch (expression){ caseClause: [defaultClause:] }	Switch (myChar) { case "A" : trace("you pressed A or a"); break; default : trace("you did not press A"); }
for (var in object) { statement(s); }	var myObject:Object = {Name:"Tara", age:27, city:"San Francisco"}; for (var prop in myObject) { trace(myObject[prop]); }
while(condition) { statement(s); }	var Cnt:Number = 0; while (Cnt < 20) { trace(Cnt); i += 3; }
do { statement(s) } while (condition)	var myVar:Number = 0; do { trace(myVar); myVar++; } while (myVar < 5);
function FName(P){ statement(s) }	function mySquared(x:Number) { return Math.pow(x, 2); }
class, interface, implement, dynamic, extend, private, public, intrinsic	Define custom class related statements

Global Functions

play, stop, nextFrame, prevFrame, gotoAndPlay, gotoAndStop, nextScene, prevScene	Main timeline movie clip play head control
loadMovie, loadMovieNum, unloadMovie, unloadMovieNum	Loads (unload) a SWF, JPEG, GIF, or PNG file from local disk or web server into a movie clip
loadVariables, loadVariablesNum	Reads data from an external file either on local disk or on web server

setInterval, clearInterval

Repeatedly execute a function (or an object).

\\===== Example=====

```
var intervalId:Number;
var count:Number = 0;
var maxCount:Number = 10;
var duration:Number = 20;

function myCallback():Void {
    trace(count);
    if(count >= maxCount) {clearInterval(intervalId);}
    count++;
}
intervalId = setInterval(this, "myCallback", duration);
```

escape, unescape

Converts the parameter to a string and encodes it in a URL-encoded format, where all nonalphanumeric characters are replaced with % hexadecimal sequences (e.g. @ to %40).

getProperty, setProperty Get (set) movie clip property

getURL

Load a web page in browser

on (event) { }

Mouse/kef event handler
 Press, release, releaseOutside, rollOut, rollOver, dragOut, dragOver, keyPress

e.g.: on (press) { startDrag(this); }

onClipEvent(movieEvent: Object) { statements; }
 Movie clip event handler. load, unload, enterFrame, mouseMove, MouseDown, MouseUp, KeyDown, KeyUp, Data.

```
onClipEvent (keyDown) {
    if (Key.getCode() == Key.RIGHT) {
        this._parent.nextFrame();
    } else if (Key.getCode() == Key.LEFT) {
        this._parent.prevFrame();
    }
}
```

startDrag, stopDrag

Makes the target movie clip draggable while the movie plays.

```
my_mc.onPress = function () {
    startDrag(this);
}
my_mc.onRelease = function() {
    stopDrag();
}
```

fsccommand

Lets the SWF file communicate with either Flash Player or the program that is hosting Flash Player, such as a web browser.

e.g. fsccommand("fullscreen", true);

isFinite, isNaN

Test number

getVersion , targetPath, Miscellaneous functions.
trace, getTimer,
removeMovieClip
duplicateMovieClip

Global Properties

_global A reference to the global object that holds the core ActionScript classes, such as String, Object, Math, and Array.

_parent Specifies or returns a reference to the movie clip or object that contains the current movie clip or object.

_root Specifies or returns a reference to the root movie clip Timeline.

This References an object or movie clip instance.

Common Classes

Object

constructor Object

Properties constructor, _proto_, prototype, _resolve,

Methods addProperty, hasOwnProperty, isPrototypeOf, isPrototypeOf, registerClass, toString, unwatch, valueOf, watch

Array

Properties CASEINSENSITIVE, DESCENDING, length, NUMERIC, RETURNINDEXEDARRAY, UNIQUESORT

Methods concat, join, pop, push, reverse, shift, slice, sort, sortOn, splice, toString, unshift.

Example var myA:Array = new Array("a","b","c");
var myN:Array = new Array(1,2,3);
var myAN:Array =myA.concat(myN);
trace(myAN.length);
// Creates array [a,b,c,1,2,3].

Date

Properties Only have properties inherited from Object.

Methods getDate, getDay, getFullYear, getHours, getMilliseconds, getMinutes, getMonth, getSeconds, getTime, getTimezoneOffset, getYear, setDate, setFullYear, setHours, setMilliseconds, setMinutes, setMonth, setSeconds, setTime, setYear, toString, valueOf, (most functions have UTC ones)

Example var my_date:Date = new Date(2004,4,25);
trace(my_date.getYear()); // output: 104
trace(my_date.getFullYear()); // output: 2004
my_date.setYear(99);
trace(my_date.getYear()); // output: 99
trace(my_date.getFullYear()); // output: 1999

Math

Properties E, LN10, LN2, LOG10E, LOG2E, PI, SORT1_2, SORT2

Methods abs, acos, asin, atan, atan2, ceil, cos, exp, floor, log, max, min, pow, random, round, sin, sqrt, tan

Example trace(Math.log(0)); // output: -Infinity
trace(Math.atan(-1)); // output: -0.785398163397448

String

Properties length

Methods charAt, charCodeAt, concat, fromCharCode, indexOf, lastIndexOf, slice, split, substr, substring, toLowerCase, toString, toUpperCase, valueOf

Example var my_str:String = new String("Hello world");
var mySubstring:String = new String();

mySubstring = my_str.substr(6,5);
trace(mySubstring); // output: world

trace (mySubstring.toUpperCase()); //WORLD

Stage

Properties align, height, scaleMode, showMenu, width

Event onResize

Methods addListener, removeListener

Example Stage.scaleMode = "noScale"
var myListener:Object = new Object();
myListener.onResize = function () {
trace("Stage size is now " + Stage.width + " by " + Stage.height);
}
Stage.addListener(myListener);

Key

Properties BACKSPACE, CAPSLOCK, CONTROL, DELETEKEY, DOWN, END, ENTER, ESCAPE, HOME, INSERT, LEFT, PGDN, PGUP, RIGHT, SHIFT, SPACE, TAB, UP, _listeners

Events onKeyDown, onKeyUp

Methods addListener, getAscii, getCode, isAccessible, isDown, isToggled, removeListener

Example var myListener:Object = new Object();
myListener.onKeyDown = function () {
trace ("You pressed a key.");
}
myListener.onKeyUp = function () {
trace ("You released a key.");
}
Key.addListener(myListener);

Mouse

Events onMouseDown, onMouseMove, onMouseUp, onMouseWheel

Methods addListener, hide, removeListener, show

Example var mouseListener:Object = new Object();

mouseListener.onMouseDown = function() {
trace("Mouse down");
};
mouseListener.onMouseMove = function() {
trace(_xmouse);
trace(_ymouse);
};
mouseListener.onMouseUp = function() {
trace("Mouse up");
};
Mouse.addListener(mouseListener);

Button

Properties _alpha, blendMode, cacheAsBitmap, enabled, filters, _focusrect, _height, _highquality, menu, _name, _parent, _quality, _rotation, scale9Grid, _soundbufTime, tabEnabled, tabIndex, _target, trackAsMenu, _url, useHandCursor, _visible, _width, _x, _xmouse, _xscale, _y, _ymouse, _yscale

Events onDragOut, onDragOver, onKeyDown, onKeyUp, onKillFocus, onPress, onRelease, onReleaseOutside, onRollOut, onRollOver, onSetFocus

Methods getDepth

Example myBtn1_btn.enabled = true;
myBtn2_btn.enabled = false;

myBtn1_btn.onRelease = function() {
trace("you clicked: " + this._name);
};
myBtn2_btn.onRelease = function() {
trace("you clicked: " + this._name);
};

TextFormat

Constructor TextFormat

Properties align, blockIndent, bold, bullet, color, font, indent, italic, kerning, leading, leftMargin, letterSpacing, rightMargin, size, tabStops, target, underline, url

Methods getTextExtent

Example var my_fmt:TextFormat = new TextFormat();
my_fmt.bold = true; my_fmt.font = "Arial";
my_fmt.size = 12; my_fmt.color = 0xFF0000;

this.createTextField("stats_txt", 5000, 10, 0, 530, 22);
stats_txt.setTextFormat(my_fmt);